

Thymeleaf – Presentation Layer Design

Pada tutorial ini, topik yang dipelajari adalah penggunaan looping, conditional, static file dan error 404 handler pada Thymeleaf.

1. Penggunaan For Looping

```
<div th:each="student, iterationStatus: ${students}">
```

Untuk melakukan iterasi dari objek kumpulan data, ekspresi yang digunakan adalah **iterationStatus** yang diletakkan pada atribut **th:each** seperti contoh diatas.

Perulangan (0,1,2,...,n) dilakukan pada object students sementara student berupa property berisi data index ke-n nya. Penaman property (pada contoh ini : student) tidak bergantung atribut apapun.

Selain indeks berurutan, Thymeleaf juga dapat membuat indeks ganjil dan genap.

```
<div th:each="student, iterationStatus: ${students}"  
th:class="${iterationStatus.odd} ? 'odd'">
```

Pertanyaan 1: Apakah value yang dihasilkan dari `${iterationStatus.odd}`?

Hasil yang dimunculkan adalah sebagai berikut.

Semua data bernilai ganjil akan memiliki tag `div class=odd` sementara yang genap tidak.

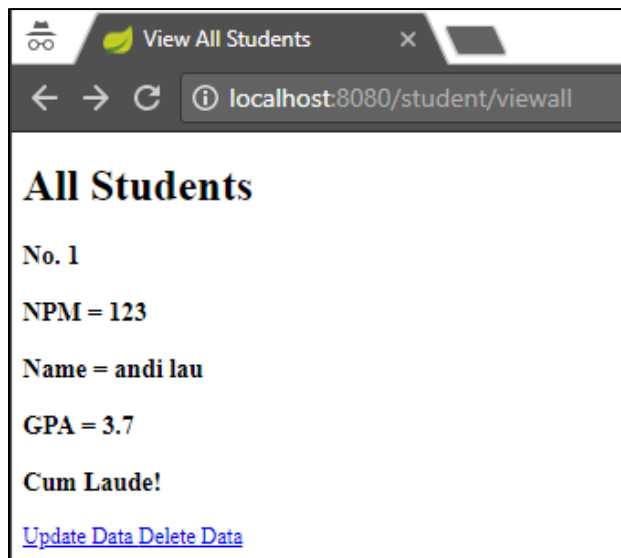
```
<h3>No. 1</h3>  
<h3>NPM = 123</h3>  
<h3>Name = andi junior</h3>  
<h3>GPA = 3.7</h3>  
<h3>Cum Laude!</h3>  
<a href="/student/update/123">Update Data </a>  
<a href="/student/delete/123">Delete Data</a><br>  
<hr/>  
</div>  
<div>  
<h3>No. 2</h3>  
<h3>NPM = 124</h3>  
<h3>Name = budi</h3>  
<h3>GPA = 3.2</h3>  
<h3>Sangat Memuaskan</h3>  
<a href="/student/update/124">Update Data </a>  
<a href="/student/delete/124">Delete Data</a><br>  
<hr/>  
</div>
```

2. Penggunaan Conditional Expression

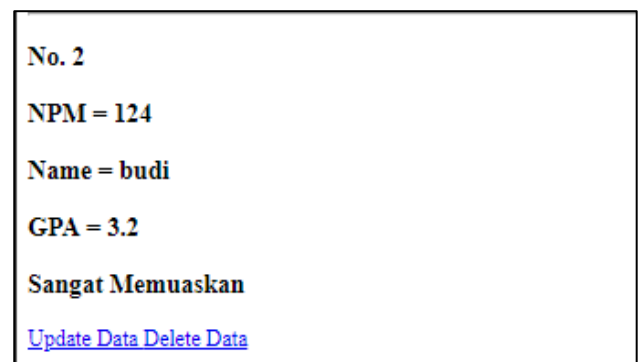
Penggunaan ini dilakukan untuk menampilkan sesuatu dengan kondisi tertentu. Untuk mencobanya, tambahkan kode baris di bawah ini pada **viewall.html**

```
<h3 th:if="{student.gpa }>=3.49}">Cum Laude!</h3>  
<h3 th:unless="{student.gpa}>=3.49}">Sangat Memuaskan</h3>
```

Apabila di run, muncul tampilan seperti di bawah ini :



Gambar 2. Tampilan dengan GPA ≥ 3.49



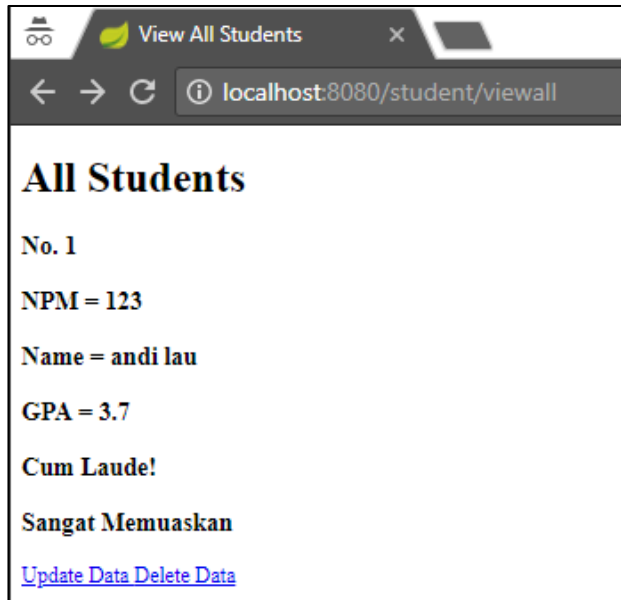
Gambar 3. Tampilan dengan GPA < 3.49

Pertanyaan 2: Mengapa condition di dalam `th:unless` sama dengan di dalam `th:if`?

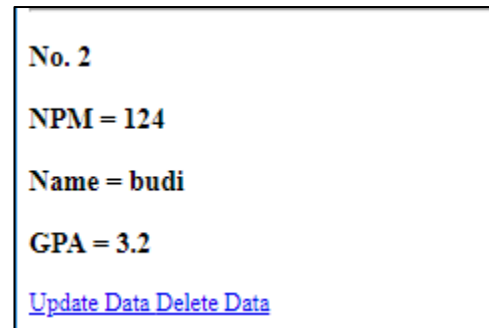
Karena nilai yang diambil adalah nilai yang bukan value pada condition `th:unless`. Nilai ≥ 3.49 . Sementara pada `th:if`. Nilai yang diambil adalah value yang ada pada kondisi `th:if`.

Pertanyaan 3: Coba ubah condition pada `th:unless={student.gpa}>=3.49}` menjadi `th:unless={student.gpa}<=3.48}`, lalu run kembali aplikasi Anda dan mengakses halaman yang sama. Apakah terjadi error? Jika iya, kenapa? Jelaskan alasan Anda. Jika tidak error, apakah hasilnya sesuai?

Page tidak error namun hasilnya tidak sesuai. Student dengan GPA ≥ 3.49 akan menampilkan Cum Laude! Sangat Memuaskan sementara GPA ≤ 3.49 tidak akan menampilkan pernyataan tersebut karena `th: unless` merujuk pada kondisi apabila GPA bukan ≤ 3.49 . Tampilannya adalah sebagai berikut :



Gambar 4. Tampilan dengan GPA ≥ 3.49



Gambar 5. Tampilan dengan GPA < 3.49

Pertanyaan 4: Tuliskan cara lain Anda dengan penulisan conditional expression yang berbeda untuk mendapatkan hasil yang sama seperti gambar di atas. Hint: Gunakan ternary operator

3. Penggunaan Static file

Contoh static file adalah file css, javascript, image, dan sebagainya. Pada spring boot dan thymeleaf, static file disimpan dan diakses dalam folder static yang sudah tersedia secara default dari pembuatan project berbasis Thymeleaf. Pada bagian ini, saya mempelajari implementasi css bootstrap.

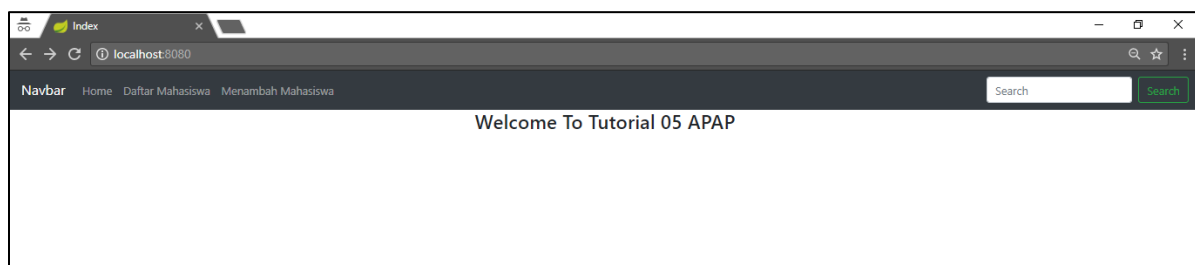
Ubah halaman **index.html** menjadi seperti di bawah ini:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title>Index</title>
    <link rel = "stylesheet" href="/css/bootstrap.min.css" />
  </head>
  <body>
    <nav class="navbar navbar-expand-Lg navbar-dark bg-dark">
      <a class="navbar-brand" href="#">Navbar</a>
      <button class="navbar-toggler" type="button"
        data-toggle="collapse"
        data-target="#navbarSupportedContent"
        aria-controls="navbarSupportedContent"
        aria-expanded="false"
        aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
```

```
        <div class="collapse navbar-collapse"
id="navbarSupportedContent">
            <ul class="navbar-nav mr-auto">
                <li class="nav-item">
                    <a class="nav-link" href="/">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link"
href="/student/viewall">Daftar Mahasiswa</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link"
href="/student/add">Menambah Mahasiswa</a>
                </li>
            </ul>
            <form class="form-inline my-2 my-lg-0">
                <input class="form-control mr-sm-2"
type="text"
placeholder="Search"
aria-label="Search" />
                <button class="btn btn-outline-success my-2
my-sm-0" type="submit">
                    Search
                </button>
            </form>
        </div>
    </nav>
    <center>
        <h3>Welcome To Tutorial 4 APAP</h3>
    </center>
</body>
</html>
```

Pada potongan program di atas, dapat disimpulkan bahwa untuk memanggil file bootstrap.min.css, tambahkan link rel="stylesheet href="lokasi_dan_nama_file_css". Setelah itu, tag css akan dapat digunakan. Folder css dan js harus dimasukkan ke dalam folder static agar dapat digunakan.

Tampilan akan menjadi seperti di bawah ini:



Gambar 6. Tampilan index setelah menggunakan bootstrap

4. Penggunaan Fragment

Fragment merupakan fitur untuk me-reuse source code halaman HTML dengan memecah halaman HTML menjadi bagian-bagian kecil dan bagian tersebut dapat digunakan oleh halaman HTML lain. Contoh komponen yang sering dipakai adalah header dan footer. Proses pembuatannya adalah sebagai berikut:

1. Folder **templates** > Buat sebuah folder **fragments**
2. Buat halaman **fragment.html** dengan isi code sebagai berikut :

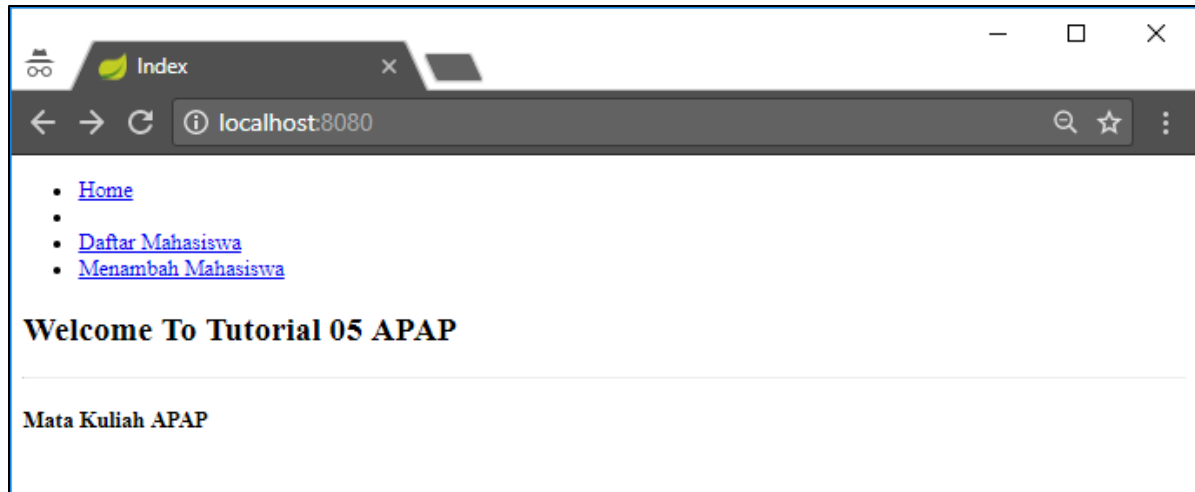
```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title>Fragment</title>
  </head>
  <body>
    <div th:fragment="header">
      <ul>
        <li><a href="/">Home</a></li>
        <li><a href="/student/viewall">Daftar
Mahasiswa</a></li>
        <li><a href="/student/add">Menambah
Mahasiswa</a></li>
      </ul>
    </div>

    <div th:fragment="footer">
      <hr/>
      <h4>Mata Kuliah APAP</h4>
    </div>
  </body>
</html>
```

3. Pada page index.html, ubah menjadi seperti di bawah ini:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title>Index</title>
  </head>
  <body>
    <div th:replace="fragments/fragment :: header"></div>
    <h2>Welcome To Tutorial 05 APAP</h2>
    <div th:replace="fragments/fragment :: footer"></div>
  </body>
</html>
```

Hasil yang muncul adalah sebagai berikut:



Gambar 7. Tampilan index menggunakan fragment

Pertanyaan 5: Apa yang dimaksud dengan **th:replace="fragments/fragment ::header"** dan **th:replace="fragments/fragment :: footer"** pada file index.html?

th:replace="fragments/fragment ::header" → berfungsi untuk menyisipkan fragment header di dalam tag div tersebut.

th:replace="fragments/fragment :: footer" → berfungsi untuk menyisipkan fragment footer di dalam tag div tersebut.

5. Penggunaan Error 404 Not Found Handler

Pembuatan handler ini berguna agar pengguna mengetahui penyebab error misalnya karena halaman tidak dapat ditemukan, dan sebagainya. Tujuannya penggunaan error handler adalah untuk meningkatkan UX value aplikasi.

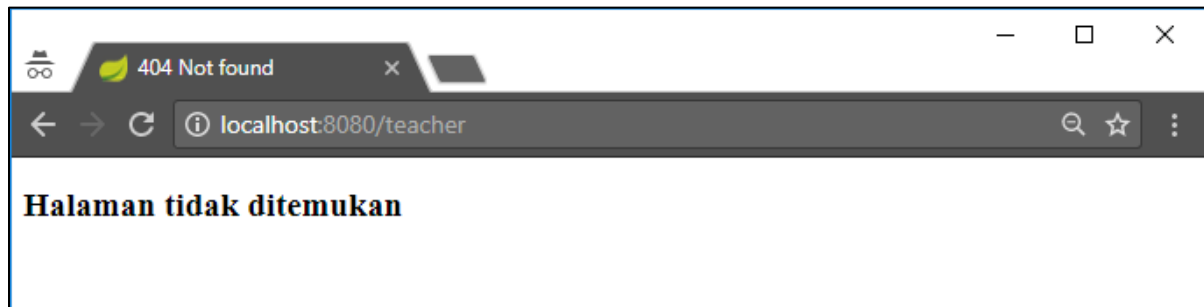
Langkah-langkahnya adalah :

1. Pada folder **template** > Buat folder error
2. Buat halaman **404.html** dengan source code berikut :

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title>404 Not found</title>
  </head>

  <body>
    <h2>Halaman tidak ditemukan</h2>
  </body>
</html>
```

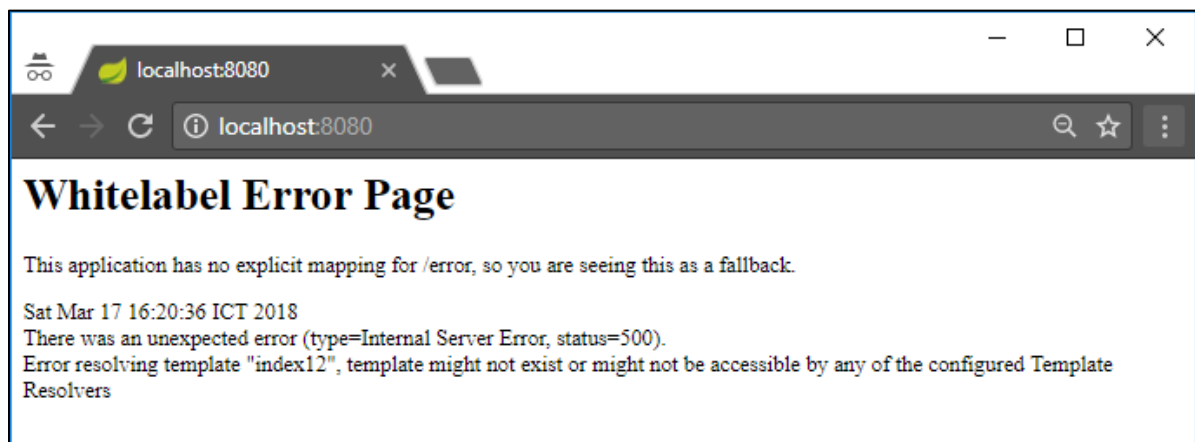
Setelah ditambahkan, apabila halaman yang diakses adalah /teacher, maka page yang muncul adalah sebagai berikut:



Gambar 7. Tampilan error 404: Not Found

Pertanyaan 6: Apa handler dengan method ini juga dapat berlaku bagi *error* lain seperti *error 500: internal server error*?

Setelah dilakukan percobaan, method tersebut tidak berlaku untuk error lain. Misalnya pada StudentController, **return “index”** diganti menjadi **“return index12”** (untuk memunculkan error 500). Maka, tampilan masih *Whitelabel Error Page* seperti di bawah ini:



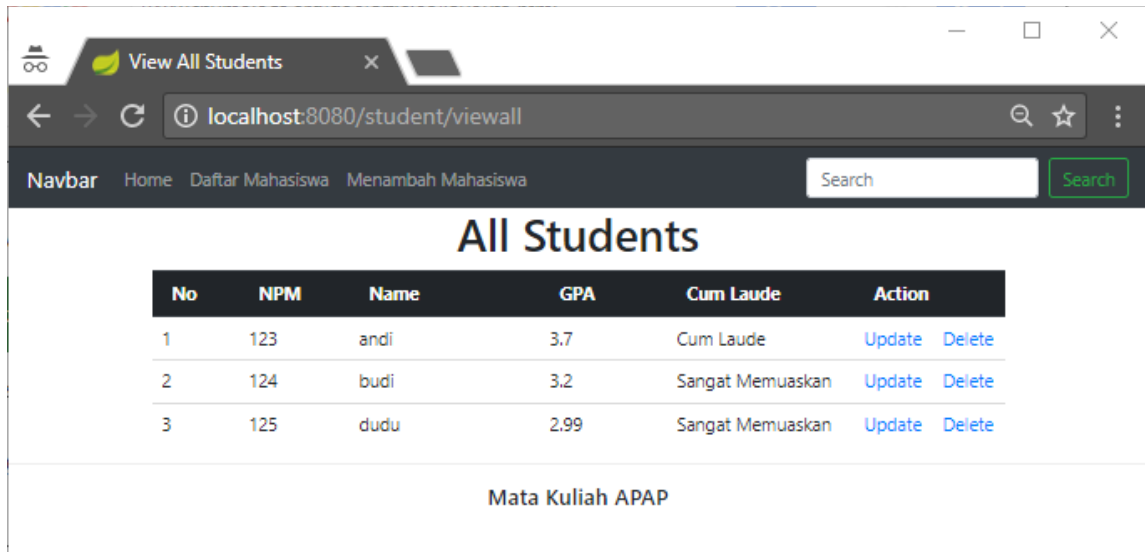
Gambar 8. Tampilan error 500: Internal server error

Latihan

1. Menggunakan plugin DataTables pada page viewall.
 1. Download dan extract DataTable di folder recourse > static
 2. Ubah viewall.html menjadi seperti di bawah ini:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title>View All Students</title>
    <link rel="stylesheet" href="/DataTables/DataTables-
1.10.16/css/jquery.dataTables.min.css" />
    <link rel="stylesheet" href="/css/bootstrap.min.css" />
    <link rel="stylesheet" type="text/css"
href="/DataTables/datatables.min.css"/>
    <script type="text/javascript"
src="/DataTables/datatables.min.js"></script>
    <script type="text/javascript"
src="/DataTables/jquery.dataTables.min.js"></script>
    <script>
      $(document).ready(function() {
        $('#example').DataTable();
      });
    </script>
  </head>
  <body>
    <h1>All Students</h1>
    <table class="table display dataTable" id="example" style="width:75%"
role="grid" >
      <thead class="thead-dark">
        <tr role="row">
          <th style="width:50px;">No</th>
          <th style="width:70px;">NPM</th>
          <th style="width:160px;">Name</th>
          <th style="width:90px;">GPA</th>
          <th style="width:147px;">Cum Laude</th>
          <th colspan="2" style="width:90px;">Action</th>
        </tr>
      </thead>
      <tbody>
        <tr th:each="student, iterationStatus: ${students}"
        <td role="row"
th:text="${iterationStatus.count}"></td>
          <td role="row" th:text="${student.npm}"></td>
          <td role="row" th:text="${student.name}"></td>
          <td role="row" th:text="${student.gpa}"></td>
          <td role="row" th:if="${student.gpa
>=3.49}">Cum Laude
          <td role="row" th:unless="${student.gpa
>=3.49}">Sangat Memuaskan
          <td role="row"><a th:href="'/student/update/' +
${student.npm}">Update Data</a>
          <td role="row"><a th:href="'/student/delete/' +
${student.npm}">Delete Data</a></td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```


Page viewall menjadi seperti berikut:



Gambar 9 . Tampilan Viewall

2. Membuat fragment header dan footer dinamis

- o Ganti file fragment.html:

```
<title th:replace="fragments/fragment ::  
header(title=${title})"></title>
```

- o Tambahkan file fragment_2.html untuk memanggil fragment yang tidak memerlukan parameter dan isi dengan source code sebelumnya.
- o Ubah title pada form-add.html, form-update.html, not-found.html, view.html, dan viewall.html menjadi :

```
<title th:replace="fragments/fragment ::  
header(title=${title})"></title>
```

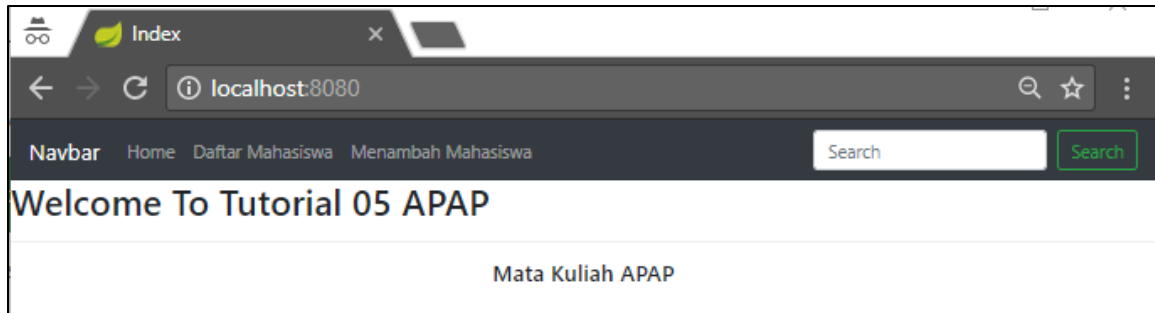
- o Sisipkan attribute model pada mapping index, view, viewall, dan update di file StudentController berikut:

```
model.addAttribute("title", "Update student");
```

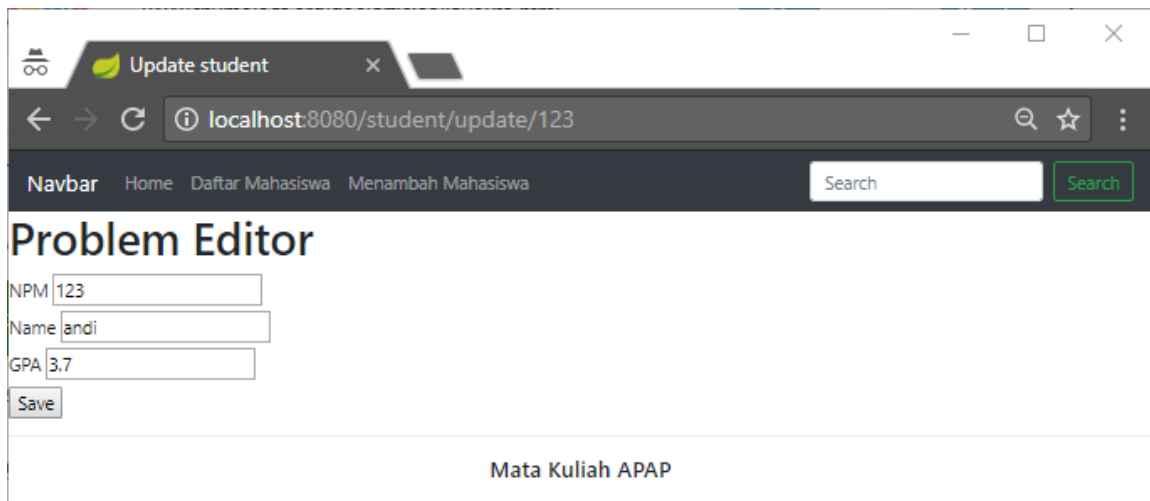
dan bagian dibawah ini untuk memanggil page not-found.html

```
model.addAttribute("title", "Student not found");
```

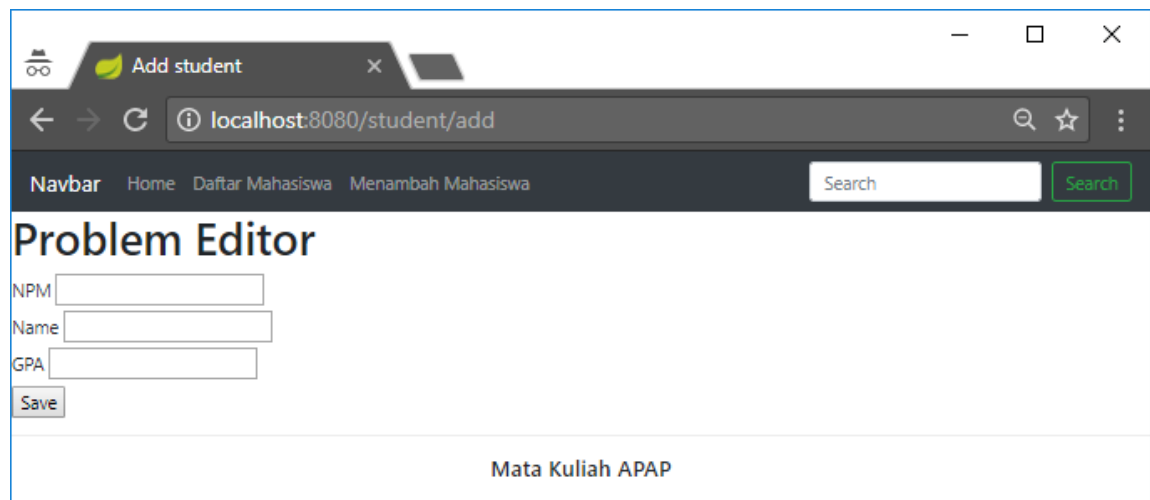
Hasilnya adalah seperti di bawah ini:



Gambar 10. Page index.html



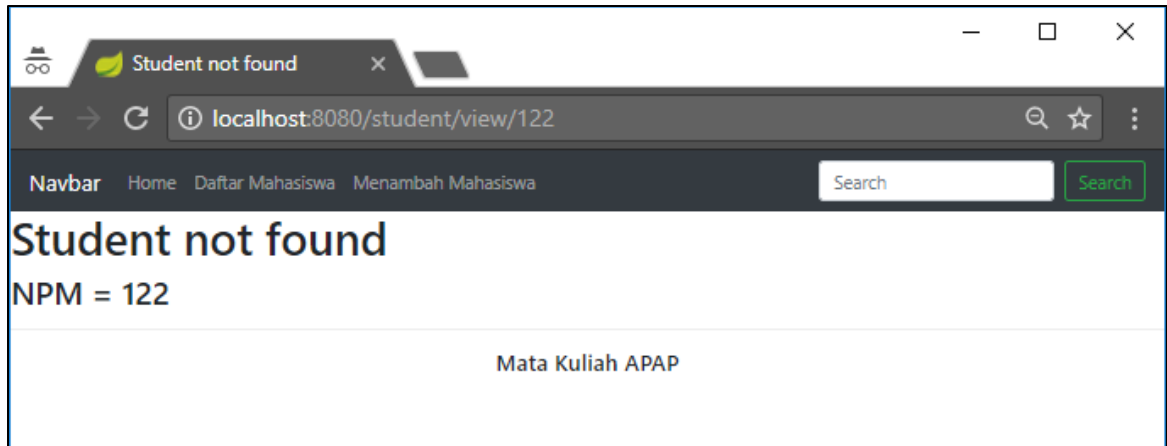
Gambar 11. Page update.html



Gambar 12. Page addStudent.html

Nama : Lidia Natalia Manik
NPM : 1706106803
Kelas : Ekstensi

Tutorial 05



Gambar 13. Page not-found.html